

The Benefit of Limited Feedback to Generation-Based Random Linear Network Coding in Wireless Broadcast

Mingchao Yu*, Parastoo Sadeghi*, and Alex Sprintson†

*Research School of Engineering, Australian National University, Canberra, Australia

†Department of Electrical and Computer Engineering, Texas A&M University, Texas, USA

Emails: {ming.yu,parastoo.sadeghi}@anu.edu.au, spalex@tamu.edu

Abstract

Random linear network coding (RLNC) is asymptotically throughput optimal in the wireless broadcast of a block of packets from a sender to a set of receivers, but suffers from heavy computational load and packet decoding delay. To mitigate this problem while maintaining good throughput, we partition the packet block into disjoint generations after broadcasting the packets uncoded once and collecting one round of feedback about receivers' packet reception state. We prove the NP-hardness of the optimal partitioning problem by using a hypergraph coloring approach, and develop an efficient heuristic algorithm for its solution. Simulations show that our algorithm outperforms existing solutions.

Index Terms

Wireless broadcast, network coding, throughput, delay, hypergraph coloring

I. INTRODUCTION

In many wireless settings there is a need to broadcast a block of data packets from one sender to multiple receivers over independent erasure channels [1]. A key problem in this context is to find an efficient broadcast scheme that allows all receivers to obtain all the data packets in the presence of erasures. It has been shown that the *random linear network coding (RLNC)* [2] technique can almost surely minimize the number of transmissions, and, in turn, maximize throughput by transmitting coded packets, which are random linear combinations of all the data packets using a sufficiently large finite

field. RLNC relies on *block-decoding*, i.e., all data packets can only be decoded when a sufficient number of coded packets have been received. However, this approach results in large packet decoding delays [1] and high computational load [3]. One possible way to reduce computational load is to use the sparse RLNC technique [4], in which the linear coefficients are more likely to be zeros. However, its packet decodings are still based on the whole block, hence the packet decoding delay is still large.

In this letter, we focus on RLNC-based broadcast schemes that partition the block of data packets into *generations* and construct the coded combinations from data packets that belong to the same generation. This technique results in a substantial reduction of the computational load. It also allows decoding of data packets that belong to the current generation before the next generation is transmitted, resulting in a significant decrease in the packet decoding delay.

While generation-based RLNC techniques are efficient in reducing the packet decoding delay and minimizing the computational load, they result in a larger number of transmissions, and, as a result, lower throughput. The main problem in this context is to design throughput-efficient generation-based RLNC schemes. This issue has been the subject of many studies, e.g., [5]–[9]. Most of the previous studies, however, focused on settings with no receiver feedback. In such settings, the sender is unaware of what packets are missing at each receiver, hence it cannot utilize this information to optimize the transmission scheme.

To the best of our knowledge, only few studies have focused on the transmission schemes with feedback. Yu et al. [10] considered the partitioning problem based on solutions borrowed from instantly decodable network coding (IDNC). However, since IDNC requires instant decodability of the coded combinations, it does not provide insight into the partitioning problem when more general coding techniques are used. Reference [11] proposes a limited feedback scheme that utilizes RLNC with block decoding. This scheme, however, still suffers from heavy computational load and large packet decoding delays.

In this letter, we study the general partitioning problem by considering a two-phase broadcast scheme with limited receiver feedback. In the first phase, each data packet in the block is transmitted once in an uncoded form. Then, the receivers feed back their packet reception state to the sender. Based on

this information, the sender partitions the packet block into generations and broadcasts a set of coded packets, where each coded packet is a combination of packet that belong to the same generation. The receivers use coded packets to recover missing data packets.

Our contributions can be summarized as follows: (i) We formulate an optimal partitioning problem and establish its NP-hardness; (ii) We propose a heuristic partitioning algorithm that strikes a balance between decoding delay and block completion time and provides a local Pareto-optimal solution. (iii) We perform an extensive simulation study that shows that our algorithm can significantly outperform the previously proposed methods in the literature.

II. PROBLEM FORMULATION

We consider a scenario in which a sender wants to broadcast a block of K data packets $\mathcal{P} = \{\mathbf{p}_k\}_{k=1}^K$ to a set of N receivers $\mathcal{R} = \{r_n\}_{n=1}^N$ through wireless channels which are subject to independent random packet erasures. In the first phase, referred to as the *systematic transmission phase*, the sender broadcasts each packet in \mathcal{P} without coding. Then, the sender collects one round of feedback from each receiver about its packet reception state. This information is represented by a binary $N \times K$ state feedback matrix (SFM) \mathbf{A} , where $\mathbf{A}(n, k) = 1$ means that r_n has missed \mathbf{p}_k and thus still wants it, and $\mathbf{A}(n, k) = 0$ means r_n has received \mathbf{p}_k . We denote by $T(k)$ the number of receivers that want \mathbf{p}_k . An example of SFM with $T(1) = 2$ is given in Fig. 1(a).

Based on the SFM, the sender then partitions \mathcal{P} into a set of M disjoint *generations*, denoted by $\mathbf{G} = \{\mathcal{G}_m\}_{m=1}^M$. We define the *rank* γ_m of a generation \mathcal{G}_m as the maximum number of packets in \mathcal{G}_m wanted by one receiver, i.e.,

$$\gamma_m = \max_{r_n \in R} \sum_{\mathbf{p}_k \in \mathcal{G}_m} \mathbf{A}(n, k). \quad (1)$$

For example, for the SFM in Fig. 1(a), generation $\{\mathbf{p}_2, \mathbf{p}_3\}$ has a rank of 2.

The sender then starts a *coded transmission phase*. In each transmission, it broadcasts a coded packet of a generation, which is a linear combination of data packets from this generation. We consider random linear network coding (RLNC) [2], which chooses the linear coefficients uniformly at random from a

finite field \mathbb{F}_q . When \mathbb{F}_q is sufficiently large, RLNC asymptotically guarantees linear independency between coded packets, so that any receiver who wants γ_m packets from \mathbf{G}_m can decode these packets after receiving γ_m coded packets of \mathbf{G}_m with high probability. Note that this decoding condition can also be achieved by deterministic codes such as maximum-distance-separation (MDS) codes and fountain codes [12], [13]. We focus on RLNC for clarity, but our techniques can be extended to other codes.

In our context, the original RLNC technique uses only one generation, i.e., $\mathbf{G}_1 = \mathcal{P}$. When it is applied, a receiver who is missing γ_1 packets from \mathcal{P} needs to receive γ_1 linearly independent coded packets of \mathcal{P} before being able to decode them, which incurs a computational complexity of $\mathcal{O}(\gamma_1^3)$ [14]. To minimize the computational load and reduce the packet decoding delay, while also maintaining a good throughput performance, we propose the following *minimum partitioning* (MP) problem:

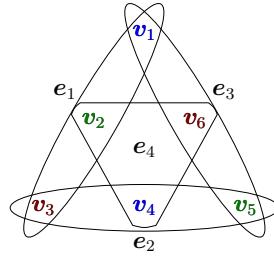
Problem MP. *Given an SFM \mathbf{A} and a constant γ , find the minimum value of M , for which there exists a partition of the packet block \mathcal{P} into M disjoint generations $\mathbf{G} = \{\mathcal{G}_m\}_{m=1}^M$ such that the rank of each generation \mathcal{G}_m does not exceed γ , i.e., $\gamma_m \leq \gamma$ for $m \in [1, M]$.*

Minimizing the total number of generations M is an important goal due to the following reasons:

- 1) The lower values of M would result in smaller number of transmissions and, as a result, higher throughput. In particular, we need at least $U \triangleq \sum_{m=1}^M \gamma_m \leq M\gamma$ transmissions in the coded transmission phase to satisfy all receivers. Due to erasures, more transmissions may be needed, but its number will depend on M , and lower values of M typically require a smaller number of transmissions.
- 2) The lower values of M reduce the overall computational load of decoding algorithm. Indeed, every receiver can decode the data packets it wants from any generation by solving a set of at most γ linear equations. This bounds the total computational load of decoding to $\mathcal{O}(M\gamma^3)$ operations.
- 3) The lower values of M reduce the average packet decoding delay (APDD) D , i.e., the average time it takes for a receiver to decode a data packet. It is calculated as:

$$D = \frac{1}{\sum_{k=1}^K T(k)} \sum_{\mathbf{A}_{(n,k)}=1} u_{n,k}, \quad (2)$$

	\mathbf{p}_1	\mathbf{p}_2	\mathbf{p}_3	\mathbf{p}_4	\mathbf{p}_5	\mathbf{p}_6
r_1	1	1	1	0	0	0
r_2	0	0	1	1	1	0
r_3	1	0	0	0	1	1
r_4	0	1	0	1	0	1

(a) State feedback matrix \mathbf{A} (b) Hypergraph \mathcal{H} Fig. 1. A state feedback matrix and its hypergraph representation. The minimum 1-regular coloring solution of \mathcal{H} requires 3 colors.

where $u_{n,k}$ is the time index in the coded transmission phase when r_n decodes \mathbf{p}_k .

Also note that the APDD of a partition is upper bounded as [10]:

$$D \leq \sum_{m=1}^M \frac{\gamma_m^2 + \gamma_m}{2}, \quad (3)$$

so that the lower values of M typically result in lower values of D .

III. THE HARDNESS OF THE MINIMUM PARTITIONING PROBLEM

In this section, we analyze the hardness of Problem MP by constructing a reduction from a hypergraph coloring problem. A hypergraph \mathcal{H} is defined by a pair $(\mathcal{V}, \mathcal{E})$, where \mathcal{V} is the set of vertices, and \mathcal{E} is the set of hyperedges. Every hyperedge $e \in \mathcal{E}$ is a subset of \mathcal{V} with size $|e| \geq 1$. A hypergraph is ω -uniform if for every hyperedge $e \in \mathcal{E}$ it holds that $|e| = \omega$. A γ -regular coloring of \mathcal{H} with M colors is a partition $\{\mathcal{V}_m\}_{m=1}^M$ of \mathcal{V} that satisfies $|\mathcal{V}_m \cap e_n| \leq \gamma$ for each $e_n \in \mathcal{E}$. The minimum number M of colors required for a γ -regular coloring of \mathcal{H} is called the γ -chromatic number of \mathcal{H} . It was shown in [15], [16] that the problem of finding the γ -chromatic number for a given hypergraph is NP-complete.

Our reduction includes the following steps. Given \mathcal{H} , for every vertex $v_k \in \mathcal{H}$ we generate a data packet \mathbf{p}_k , and for every hyperedge e_n we generate a receiver r_n who wants the data packets that correspond to the vertices that are incident to e_n . For example, the SFM in Fig. 1(a) can be generated from the hypergraph depicted in Fig. 1(b), with 6 data packets/vertices and 4 receivers/hyperedges.

Let $\{\mathcal{V}_m\}_{m=1}^M$ be a γ -regular coloring solution of the instance \mathcal{H} of the γ -regular coloring problem. Then, let $\mathbf{G} = \{\mathcal{G}_m\}_{m=1}^M$ be a partition of \mathcal{P} such that for $m = 1, \dots, M$, a generation \mathcal{G}_m includes packets that correspond to vertices in \mathcal{V}_m . It is easy to verify that \mathbf{G} is a valid solution to Problem MP with M generations such that the rank of each generation is bounded by γ .

Similarly, for any solution $\mathbf{G} = \{\mathcal{G}_m\}_{m=1}^M$ of Problem MP, we can construct a γ -regular coloring solution $\{\mathcal{V}_m\}_{m=1}^M$ with M colors, in which each set \mathcal{V}_m includes vertices that correspond to packets in \mathcal{G}_m . Thus, an optimal algorithm for Problem MP can be used to find the γ -chromatic number of a hypergraph, which is an NP-complete problem. We summarize our results in the following theorem:

Theorem 1. *Problem MP is NP-complete.*

We note that our reduction can also be used to establish the hardness of several special cases of our problem based on other results in graph theory. For example, it is an NP-complete problem to find 2-regular coloring with two colors for a 3-uniform hypergraph [16], [17]. This implies that, in the special case in which every receiver wants $\omega = 3$ data packets, it is NP-complete to find a partition of size $M = 2$ with the rank of each generation being bounded by $\gamma = 2$. In addition, it is NP-complete to find the chromatic number of hypergraphs in the special case of 1-regular coloring [18]. This implies that our problem is intractable even for the special case of $\gamma = 1$, in which every generation can include at most one missing packet for any receiver [10]. Thus, by using our technique we can recover the hardness result of instantly decodable network coding established in [19] using a different approach.

IV. PARTITION ALGORITHM

In this section, we propose a heuristic algorithm for Problem MP. In addition to aiming to minimizing the number of partitions, our algorithm also orders the partitions in such a way that popular packets (i.e., packets with large values of $T(k)$) are assigned to early generations. The partition ordering satisfies the *irreducibility* property, i.e., no data packet from a generation \mathcal{G}_m can be moved to an earlier generation \mathcal{G}_n ($n < m$) without increasing its rank [20]. The irreducibility property, in turn, implies that the proposed algorithm achieves local *Pareto-optimal* trade-off between throughput and packet decoding delay delay.

Note that another approach for Problem MP is to find the corresponding instance of the hypergraph coloring problem and then use existing hypergraph coloring algorithms. However, existing approximation algorithms for the hypergraph coloring problem mainly focus on the special case of uniform hypergraphs (e.g., [16]) and do not consider other aspects of the solution mentioned above.

The detailed description of the algorithm is provided in Algorithm 1. The algorithm receives as

Algorithm 1 Feedback-assisted minimum partition

```

Initialize: SFM  $\mathbf{A}$ , generation rank  $\gamma$ , a counter  $m = 0$ 
while  $\mathcal{P}$  is nonempty do
     $m \leftarrow m + 1$ , create an empty generation  $\mathcal{G}_m$ ;
    while there exists a packet in  $\mathcal{P}$  that would not increase the current rank of  $\mathcal{G}_m$ ,  $\gamma_m$ , to  $\gamma + 1$  do
        if there exists a packet in  $\mathcal{P}$  that would not increase  $\gamma_m$  by one then
            add the most popular packet that satisfies this condition to  $\mathcal{G}_m$ , and then remove it from  $\mathcal{P}$ ;
        else
            add the most popular packet in  $\mathcal{P}$  to  $\mathcal{G}_m$ , and then remove it from  $\mathcal{P}$ . Increase  $\gamma_m$  by one;
        end if
    end while
end while

```

inputs a state feedback matrix \mathbf{A} and the upper bound γ of the rank of each generation. The algorithm iteratively creates generations and fills them with packets. At each step, the algorithm searches for a packet that can be added to the current generation without increasing its rank. If such a packet exists, it is added to the generation. If such a packet does not exist and the rank of the current generation is less than γ , then the most popular packet in \mathcal{P} is added to the current generation. A packet that is added to the current generation is removed from \mathcal{P} . If the size of the current generation is γ and no packet can be added to it without increasing its rank, then a new generation is formed. The algorithm starts with empty generation \mathcal{G}_1 and terminates when set \mathcal{P} becomes empty.

V. NUMERICAL RESULTS

In this section, we compare the performance of our algorithm with alternative solutions. Note that once the packet block is partitioned into generations, there are several different ways to schedule transmissions in the coded transmission phase to satisfy all the receivers. In our experiments, we used the following

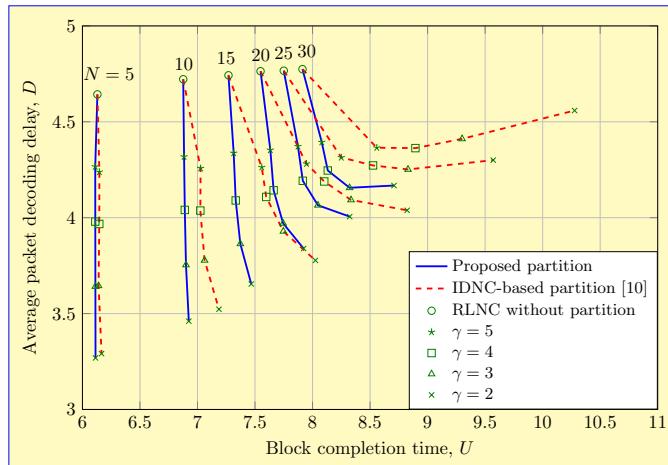


Fig. 2. The throughput-delay tradeoffs of the proposed and IDNC-based partition algorithms for different numbers of receivers N .

strategy. The sender broadcasts the generations in a round-robin manner until all receivers are satisfied and feedback ACKs. In every transmission round γ_m coded packets of each generation \mathcal{G}_m are broadcast. We note that several more advanced strategies may also be used. For example, the sender can send more than γ_m coded packets of each \mathcal{G}_m in the first round to increase the probability of block completion after this round, or it can merge generations together with the help of extra feedback to reduce redundant packets [10], [11].

Fig. 2 compares the performance of the partition solutions produced by our algorithm with the IDNC-based algorithm presented in [10] in terms of throughput-delay performance, measured by the block completion time U and average packet decoding delay D , respectively. In our experiments, the packet block size is $K = 20$ and the packet erasures in the systematic transmission phase are random i.i.d. random variables with a probability of $P_e = 0.2$. In this figure only, we assume that the coded transmissions are erasure-free. Our experimental results show that the proposed algorithm always outperforms the IDNC-based one in terms of U . The improvement becomes more significant with increasing number of receivers. The D of the two algorithms is similar when γ is large, but our algorithm outperforms the IDNC-based one with increasing number of receivers when γ is small.

We also evaluate the gain achieved by our algorithm, enabled by the feedback collected after the systematic transmission phase. To this end, we compare the throughput and decoding delay of the coded transmission phase when our algorithm and the *blind* partition approach are applied, respectively. With the blind partition approach, we sequentially segment the packet block into M generations, each carrying $\frac{K}{M}$ data packets, where M is set to be the number of generation produced by our algorithm. If the blind partition is applied, only one coded packet of every generation is broadcast in each round, as the sender is unaware of the generation ranks.

In our experiments, the packet block size is $K = 20$, and the packet erasures are random i.i.d. random variables with a probability of $P_e = 0.2$ in both the systematic and coded transmission phases. The simulation results are shown in Fig. 3. We observe that by collecting only one round of feedback, our algorithm can reduce the block completion time by 30% and the average packet decoding delay by 40%, compared to the blind partition. The improvement decreases with increasing γ , because both

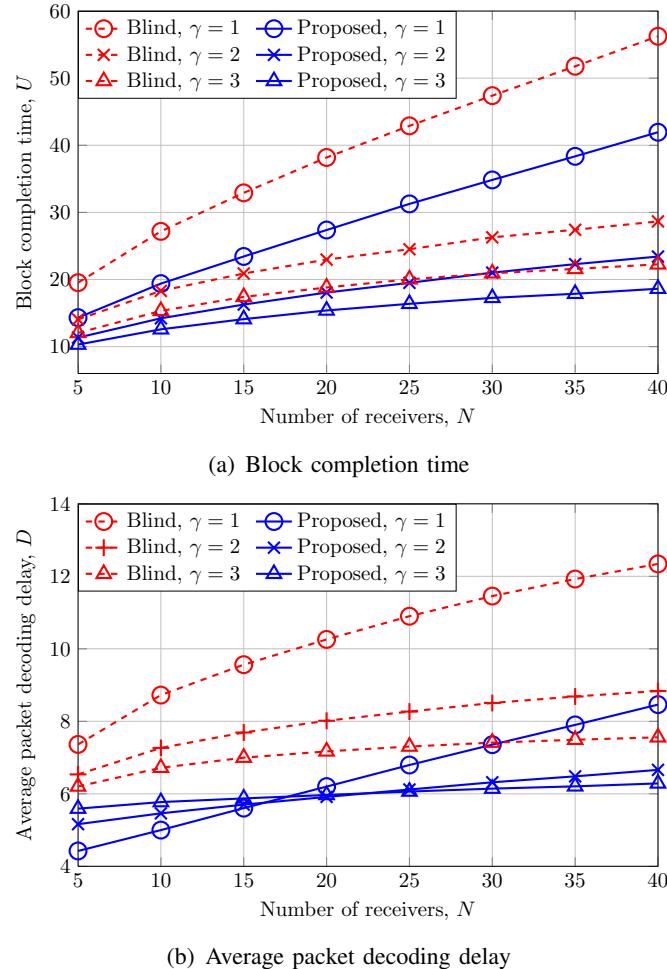


Fig. 3. Performance improvements with one round of feedback.

techniques will converge to the original RLNC.

VI. CONCLUSION

In this letter, we showed that feedback-assisted partitioning can bring a better tradeoff in terms of throughput, computational load, and packet decoding delay in RLNC-based wireless broadcast. We established the NP-completeness of optimal partition problem and presented a heuristic partition algorithm that achieves local Pareto-optimality. Our extensive simulations show that the proposed algorithm outperforms previously proposed solutions. As a future work, we are interested in developing generation-based approaches that use deterministic coding techniques, such as MDS codes and fountain codes.

We are also interested in designing a comprehensive solution to the wireless broadcast problem that includes partitioning, scheduling, and an optimal feedback mechanism.

We believe that the connection between wireless broadcast and hypergraph coloring established in this letter will provide new insights into other network settings such as cooperative data exchange problem.

REFERENCES

- [1] L. Keller, E. Drinea, and C. Fragouli, “Online broadcasting with network coding,” in *Proc. Workshop on Network Coding, Theory and Applications (NetCod)*, 2008, pp. 1–6.
- [2] T. Ho, M. Médard, R. Koetter, D. Karger, M. Effros, J. Shi, and B. Leong, “A random linear network coding approach to multicast,” *IEEE Trans. Inf. Theory*, vol. 52, no. 10, pp. 4413–4430, 2006.
- [3] J. Heide, M. V. Pedersen, F. H. Fitzek, and M. Médard, “On code parameters and coding vector representation for practical RLNC,” in *Proc. IEEE Int. Conf. Communications. (ICC)*, 2011.
- [4] C. Fragouli, J. Widmer, and J. Le Boudec, “Efficient broadcasting using network coding,” *IEEE/ACM Trans. Netw.*, vol. 16, no. 2, pp. 450–463, 2008.
- [5] P. Maymounkov, N. J. Harvey, and D. S. Lun, “Methods for efficient network coding,” in *Proc. 44-th Allerton Conference*, 2006.
- [6] D. Silva, W. Zeng, and F. R. Kschischang, “Sparse network coding with overlapping classes,” in *Proc. 5th Workshop on Network Coding, Theory, and Applications (NetCod)*, 2009.
- [7] A. Heidarzadeh and A. H. Banihashemi, “Overlapped chunked network coding,” in *Proc. IEEE Information Theory Workshop (ITW)*, 2010.
- [8] Y. Li, P. Vingelmann, M. V. Pedersen, and E. Soljanin, “Round-robin streaming with generations,” in *Proc. IEEE Int. Symp. Network Coding (NetCod)*, 2012.
- [9] G. Joshi and E. Soljanin, “Round-robin overlapping generations coding for fast content download,” in *Proc. IEEE Int. Symp. Information Theory (ISIT)*, 2013.
- [10] M. Yu, P. Sadeghi, and N. Aboutorab, “From instantly decodable to random linear network coding,” *IEEE Trans. Commun.*, vol. 62, no. 11, pp. 3943–3955, Oct. 2014.
- [11] A. Rezaee, L. Zeger, and M. Médard, “Speeding multicast by acknowledgment reduction technique (SMART),” in *Proc. IEEE Global Telecomm. Conf. (GLOBECOM)*, 2011.
- [12] A. Shokrollahi, “Raptor codes,” *IEEE Trans. Inf. Theory*, vol. 52, no. 6, pp. 2551–2567, 2006.
- [13] M. Luby, “LT codes,” in *Proc. IEEE Symp. Foundations of Computer Science (FOCS)*, 2002, pp. 271–280.
- [14] J. Heide, M. V. Pedersen, F. H. P. Fitzek, and T. Larsen, “Network coding for mobile devices - systematic binary random rateless codes,” in *Proc. IEEE Int. Conf. Communications (ICC) workshop*, 2009.
- [15] M. Krivelevich and B. Sudakov, “The chromatic numbers of random hypergraphs,” *Random Structures and Algorithms*, vol. 12, no. 4, pp. 381–403, 1998.
- [16] ——, “Approximate coloring of uniform hypergraphs,” *Journal of Algorithms*, vol. 49, no. 1, pp. 2–12, 2003.
- [17] Lovász, “Coverings and colorings of hypergraphs,” in *Proc. 4th S.E. Conf. on Combinatiroics, Graph Theory and Computing*, 1973.
- [18] G. Agnarsson and M. M. Halldórsson, “Strong colorings of hypergraphs,” in *Approximation and Online Algorithms*. Springer, 2005, pp. 253–266.
- [19] E. Rozner, A. P. Iyer, Y. Mehta, L. Qiu, and M. Jafry, “ER: Efficient retransmission scheme for wireless LANs,” in *Proc. ACM CoNEXT*, 2007.
- [20] P. Sadeghi, M. Yu, and N. Aboutorab, “On throughput-delay tradeoff of network coding for wireless communications,” in *Proc. IEEE Int. Symp. Information Theory and its Applications (ISITA)*, 2014, pp. 689–693.